Altera MAX+PLUS II Overview

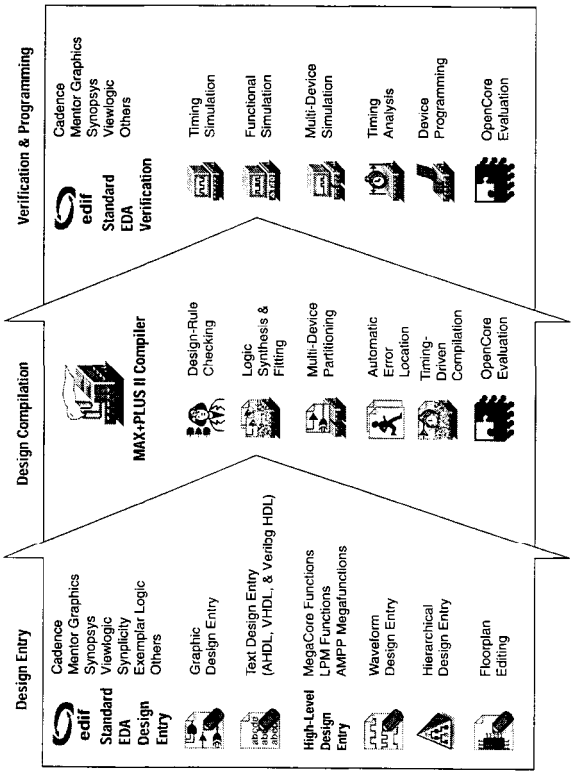

Figure from Altera technical literature

Using the MAX+PLUS II Software on vlsi.mcs.kent.edu

- Everyone in this class has access to "vlsi"
  - HP C180, 384MB of memory
- To run MAX+PLUS II:
  - Either type this line before running MAX+PLUS II, or add it to your ".cshrc" file so that it executes every time you log in:

    setenv LM_LICENSE_FILE /local/opt maxplus2/adm/license.altera
  - Run MAX+PLUS II by typing:

    /local/opt/maxplus2/bin/maxplus2
  - The first time it runs, it will copy the initialization file "maxplus2.ini" to your home directory
  - If you telnet from another workstation's console, you must tell it to give vlsi access to its display ("xhost +vlsi"), and set your DISPLAY variable on vlsi to point to that console ("setenv DISPLAY *machine*:0.0")

Using Predesigned Components in MAX+PLUS II

- Various components available
  - "Old-style macrofunctions" = 74-style
    - Familiar, but not recommended since they aren't as optimized as components below
  - Primitives
    - Gates, flip-flops, buffers
  - Standardized LPM (Library of Parameterized Modules) functions
    - Gates, arithmetic, & storage components
  - Complex predesigned "megafunctions"
    - Controllers, DSPs, communications…
    - Others available from Altera's partners
      - Use non-savable "preview" to evaluate
        - » Buy post-synthesis netlist for $
        - » Buy HDL source for $$$
- For documentation on these components, see MAX+PLUS II online help, or info on Altera's web site (link on class web page) or the Altera Digital Library CDROM

Getting Help in MAX+PLUS II

- Help menu provides basic documentation
  - AHDL, VHDL, Verilog
  - Megafunctions / LPM (LPM grouped by function, then megafunctions)
  - Old-Style Macrofunctions (grouped by function, also available by number)
  - Primitives (grouped by function)
  - Devices and adapters (FLEX 10K, etc.)
  - Messages, glossary, info about this release, how to use help
- Help syntax
  - Most links are shown in green text
    - Underlined links jump to new help topics
    - Dotted-underlined links pop up a glossary entry
  - Blue links pop up an example, list of shortcuts, or illustration

Getting Help in MAX+PLUS II (cont.)

- Help menu also provides application-specific documentation (e.g., Graphics Editor Help) in a sub-menu
  - Introduction (how to get started)
  - Basic Tools (inputs/outputs, tools, toolbars, components, examples)
  - Commands (details on every command in the application, examples, illustrations)
  - Procedures (step-by-step instructions on how to perform various tasks in the application)
  - Golden Rules (essential tips and rules for using the application)
  - Shortcuts (keyboard, mouse button, and toolbar shortcuts for commands in the application)
  - Messages (explanation of error messages for the application)

Design Entry (cont.)

- Logic synthesis
  - Allows design entry at a “higher” level of abstraction — important for complex designs
  - Input:
    - State machines
    - Boolean equations

    using a Hardware Description Language (HDL) such as
    - VHDL
    - Verilog
    - AHDL, etc. (vendor-specific)
  - Input:
    - Timing diagrams (waveforms)

    using a waveform editor
  - Then the logic synthesis tools will generate the schematic

The “Chiptrip” Tutorial Example, Revisited

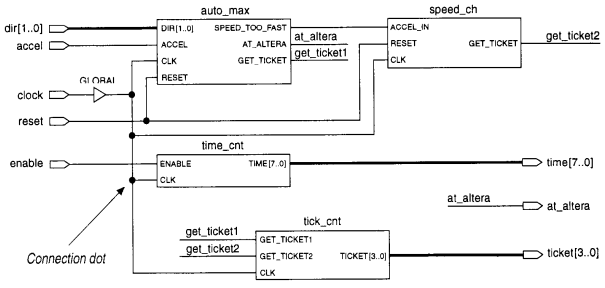

Figure from Altera technical literature

- Simulates an auto driving around town
  - auto_max — AHDL state machine that keeps track of location of auto and acceleration at that point in time, gives ticket if you accelerate on small roads
  - speed_ch — waveform state machine that gives ticket if you accelerate for a second time
  - tick_cnt — counter that counts tickets
  - time_cnt — AHDL counter that keeps track of time taken to reach Altera

Text Editor

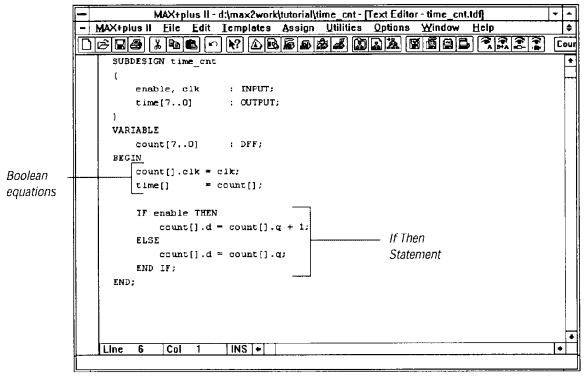

Figure from Altera technical literature

- Provides AHDL, VHDL, Verilog templates
- Syntax coloring, drag-and-drop editing, ability to find matching delimiters
- Compiler / simulator automatically locates and highlights errors
- Context-sensitive help (arrow-? button)

Sample VHDL Code for Logic Synthesis

- Part of the code describing a variable-width shift register:

```
architecture Behave of ShiftN is
    begin Shift: process (CLR, CLK)
        if CLR = '1' then
            St := (others => '0');
            Q <= St after TCQ;
        elsif CLK'EVENT and CLK='1' then
            if LD = '1' then
                St := (others => '0');
                St(InB) := D;
                Q <= St after TLQ;
            elsif SH = '1' then
                case DIR is
                when '0' => St :=
                    '0' & St(St'LEFT downto 1);
                when '1' => St :=
                    St(St'LEFT-1 downto 0) & '0';
                end case;
                Q <= St after TSQ;
            end if;
        end if;
    end process;
```


Sample VHDL Code for Logic Synthesis

- Part of the code describing a finite state machine to control a multiplier datapath:

```
process (CLK, Reset) begin
    if Reset = '1' then State <= E;
    elsif CLK'EVENT and CLK = '1' then
        case State is
        when I => State <= C;
        when C =>
            if LSB = '1' then State <= A;
            elsif Stop = '0' then State <= S;
            else State <= E;
            end if;
        when A => State <= S;
        when S => State <= C;
        when E =>
            if Start = '1' then State <= I; end if;
        end case;
    end if;
 end process;
 end;
```


Design Using Altera's MAX+PLUS II

- Follow along in Hamblen Sections 1.5 and 1.6, end of Chapter 1
  - VHDL design entry
  - Entity versus architecture
- Verilog
- Symbols and hierarchy
- Functional simulation